

РАЗРАБОТКА КОМПЬЮТЕРНОЙ 2D ИГРЫ

Плотников М.С.

Тамбов, Технический колледж ФГБОУ ВО «Тамбовский государственный технический университет», 4 курс

Научный руководитель: Мосягина Надежда Геннадьевна, к.т.н. преподаватель, г. Тамбов, Технический колледж ФГБОУ ВО «Тамбовский государственный технический университет»

Основной жанр игры «Mad Fang» – аркадный платформер. Игра разрабатывалась в развлекательных целях. Целевая аудитория игры – это дети от 9 лет, которым интересен игровой процесс, где можно соревноваться с друзьями на скорость прохождения и количество очков, интересные враги, много ловушек и потайных комнат.

Главный герой игры – лис по имени Fang. Его основная цель состоит в том, чтобы выжить в игровом мире, населённом монстрами, которые всячески пытаются помешать ему.

Игра рассчитана на одного игрока (singleplayer), но наличие интересного игрового процесса, мелодичных звуков и приятной атмосферы не даёт заскучать игроку, заставляя его возвращаться в игру снова и снова, а понятная инструкция, различные подсказки и удобное управление не вызывают сложностей в изучении правил игры.

Управление героем игры и игровым полем осуществляется с помощью меню, рис. 2.

Суть игры состоит в прохождении увлекательных уровней, где главный герой – лис побеждает врагов, которые пытаются помешать дойти до конца. Основной чертой игрового процесса является перемещение (прыжки)

по платформам и сбор предметов, обычно необходимых для завершения уровня (сердечек для восстановления «жизней» и тортов, играющих роль очков) и использование ортогональной карты, в которой использована двумерная графика, т.е «камера сбоку».

Многочисленные противники (называемые «врагами»), обладают примитивным интеллектом, стараются ударить главного героя, тем самым нанося ему урон и отнимая часть жизней, а впоследствии, и полностью убивая игрока. Противник может быть нейтрализован прыжком ему на голову. Уровни изобилуют различными секретами, например невидимыми стенами и движущимися платформами, с которых можно упасть в места из которых сложно выбраться, тем самым поддерживая интерес игрока.

В качестве средства разработки игры использовался язык программирования C++, так как он обладает необходимыми возможностями для реализации элементов игры за счет использования объектно-ориентированного подхода. Средой разработки было решено выбрать Microsoft Visual Studio Community 2015, так как этот продукт удобен и свободно распространяется [1-2].

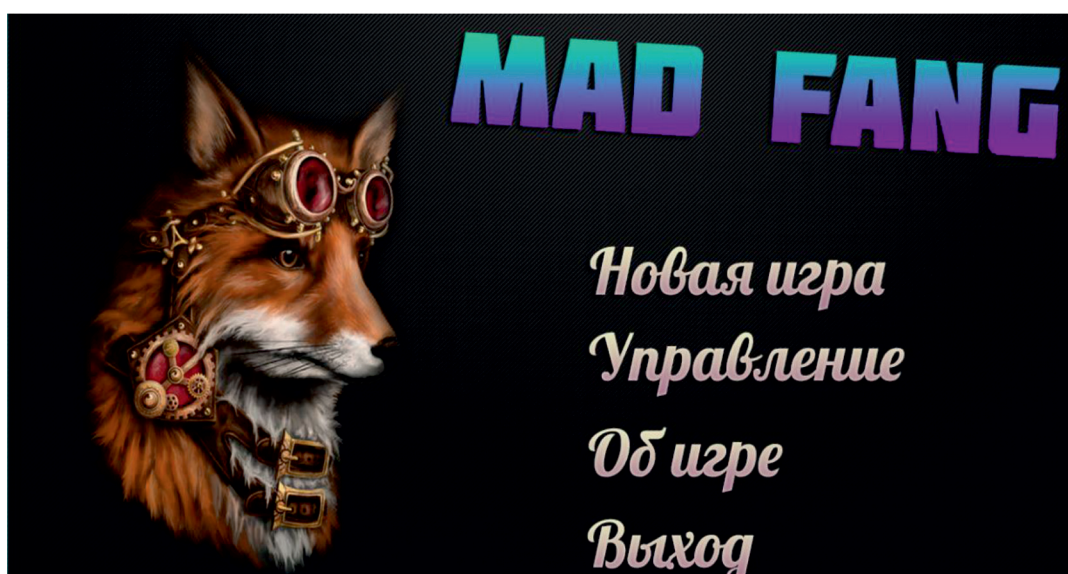


Рис. 1. Главное меню игры

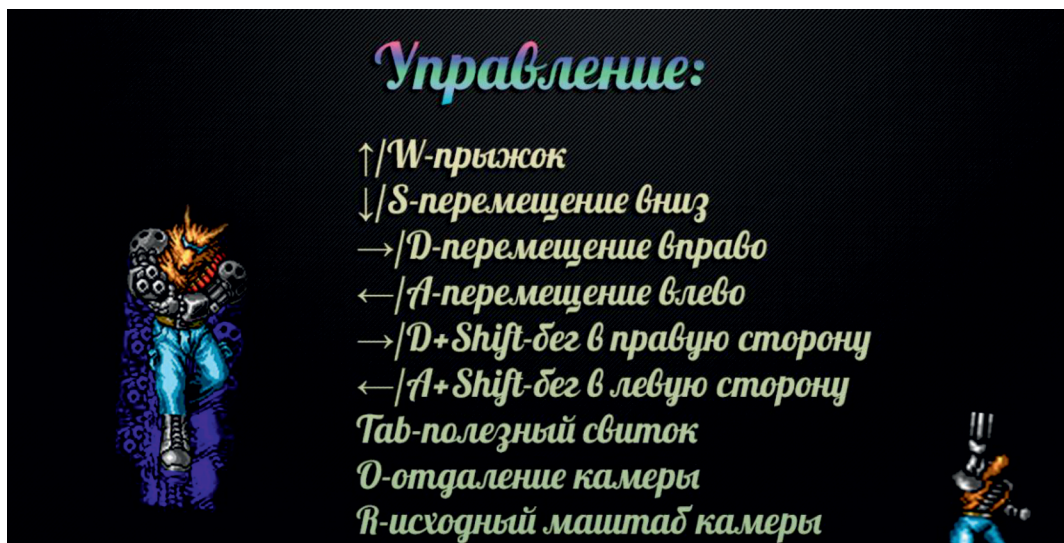


Рис. 2. Управление героем



Рис. 3. Игровое пространство, первый уровень

Для реализации игры была использована библиотека SFML (Simple and Fast Multimedia Library) – свободная мультимедийная библиотека, специализированная для создания игр, включающая в себя средства создания окон, графики и звуков, а так же библиотеки для работы с Tile map

editor-кроссплатформенный открытый редактор тайловых карт для игр. Он позволяет создавать и редактировать карты для 2-мерных игр [3, 4].

В качестве иллюстрации реализации стратегий игры приведен пример кода (взаимодействие персонажа с объектами карты):

```
void checkCollisionWithMap(float Dx, float Dy, float time) //функция проверки столкновений с картой
{
    moveTimer += time;
    for (int i = 0; i < obj.size(); i++) //проходим по объектам
        if (getRect().intersects(obj[i].rect)) //проверяем пересечение с объектом
            if (obj[i].name == "solid") //если препятствие
```

```

        {if (Dy > 0) { y = obj[i].rect.top - h; dy = 0; onGround = true; }
        if (Dy < 0) { y = obj[i].rect.top + obj[i].rect.height; dy = 0; }
        if (Dx > 0) { x = obj[i].rect.left - w; }
        if (Dx < 0) { x = obj[i].rect.left + obj[i].rect.width; }}

    if (obj[i].name == "kaktys") //если взяли кактус,то переменная health=health-15;
        {if (moveTimer > 5000) {health -= 15; moveTimer = 0; }}
        if (obj[i].name == "tort") //если взяли торт, переменная
playerScore=playerScore+1
        {playerScore++;
obj[i].name = "pusto";}
        if (obj[i].name == "heals")
        {health += 30; //если взяли сердечко,то переменная health=health+30;
        if (health > 100) health = 100; else obj[i].name = "pusto";
        }if (obj[i].name == "doorw") chngl = 1;}}

```

Функция принимает значения положений игрока по его координатам, а также переменную времени для таймера. Если координаты игрока пересекают объект, например стену, то в эту текстуру его не пускают, отталкивая от стены в противоположном направлении.

В следующем примере создаётся главный класс, в котором будут присутствовать переменные, которые соответствуют и персонажу и врагам – «жизнь», положение и т.д. Далее создаётся дочерний класс Игрок, который принимает изначальное положение на карте, количество его «жизни» и рисуется сама модель:

```

class Entity { //Главный класс
public:
    std::vector<Object> obj; //вектор объектов карты
    float dx, dy, x, y, speed, moveTimer;
    int w, h, health;
    bool life, isMove, onGround;
    Texture texture;
    Sprite sprite;
    String name;
    Entity(Image &image, String Name, float X, float Y, int W, int H) {
        x = X; y = Y; w = W; h = H; name = Name; moveTimer = 0;
        speed = 0; health = 100; dx = 0; dy = 0;
        life = true; onGround = false; isMove = false;
        texture.loadFromImage(image);
        sprite.setTexture(texture);
        sprite.setOrigin(w / 2, h / 2);
    }
    FloatRect getRect() { //функция получения прямоугольника
        return FloatRect(x, y, w, h); функция проверки столкновений
    }
    virtual void update(float time) = 0;
};

class Player :public Entity { //Класс Игрок
public:
    float CF = 0;
    bool isShoot, chngl;
    enum { left, right, up, down, jump, stay} state;
    int playerScore, health;
    Player(Image &image, String Name, Level &lev, float X, float Y, int W, int H)
:Entity(image, Name, X, Y, W, H) {
        playerScore = 0; state = stay; obj = lev.GetAllObjects(); health = 100;//для взаи-
модействия персонажа с картой
        if (name == "Player1") {координаты модельки игрока

```

Особое внимание уделялось анимации персонажа. Естественно надо заставить его реагировать на действия игрока, в частности заставить его перемещаться. Но персонаж при движении не может быть статичным, поэтому была добавлена простая анимация движений, когда герой поднимает колени при беге: при нажатии определённой кнопки координаты нашего персонажа меняются и текстура(тайл) сдвигается на определённое количество пикселей. Также было реализовано отдаление камеры, возвращение её в исходное положение.



Рис. 4. Анимация персонажа

Оставалось не так много – добавить звуки и музыку фона с помощью возможностей SFML и класс движущихся платформ с такой логикой, чтобы находя преграду, она двигалась в противоположном направлении. Также создать текстовый счётчик «жизней» и на основе его реализовать «полосу здоровья», которая уменьшается при использовании «жизни».

Проект Игра «Mad Fang» был представлен на IX областном конкурсе информационных и компьютерных технологий «Компьютер – XXI век» Тамбовской области и удостоен диплома 3-й степени.

Работы над проектом позволила глубже изучить возможности языка программирования C++ и библиотеки SFML, приобрести практические навыки разработки игровых приложений.

Список литературы

1. Герберг Шилдт С++. Базовый курс, 2015. – 624 с.
2. Николаев Е.И. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие / Николаев Е.И. – Электрон. текстовые данные. – Ставрополь: Северо-Кавказский федеральный университет, 2015. – 225 с. – Режим доступа: <http://www.iprbookshop.ru/62967.html>.
3. SFML – Библиотеки разработки игр [Электронный ресурс]. – Режим доступа: <https://progressor-blog.ru/category/sfml>.
4. Журнал «Информационные технологии моделирования и управления» [Электронный ресурс]: – Режим доступа: <http://www.iprbookshop.ru/51662.html>.